

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-40099

(P2000-40099A)

(43) 公開日 平成12年2月8日(2000.2.8)

(51) Int.Cl.

G 0 6 F 17/60

識別記号

F I

C 0 6 F 15/21

テマコード(参考)

L 5 B 0 4 9

審査請求 未請求 請求項の数10 O L (全 13 頁)

(21) 出願番号

特願平10-207880

(22) 出願日

平成10年7月23日(1998.7.23)

(71) 出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72) 発明者 守屋 哲朗

神奈川県川崎市幸区柳町70番地 株式会社

東芝柳町工場内

(72) 発明者 成松 克己

神奈川県川崎市幸区柳町70番地 株式会社

東芝柳町工場内

(74) 代理人 100081961

弁理士 木内 光春

Fターム(参考) 5B049 BB07 CC21 CC32 DD01 DD05

EE01 EE05 FF02 FF03 FF07

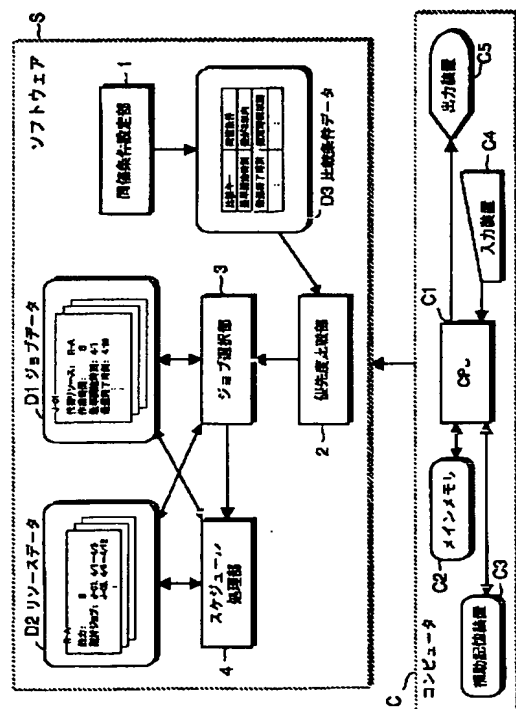
FF09

(54) 【発明の名称】 スケジュール作成装置及び方法、ジョブの選択方法並びにスケジュール作成用ソフトウェアを記録した記録媒体

(57) 【要約】

【課題】 ユーザの意図を直接反映したスケジューリングの技術を提供する。

【解決手段】 比較条件データD3は、ジョブの比較を行うときの比較条件、すなわち、ジョブごとに複数存在する属性値のうち、どれとどれをどのような優先順位で比較キーにするかと、比較キーのうち少なくとも1つについての同値条件と、を含んでいる。ジョブ選択部3は、選択対象となっているジョブ、すなわちまだリソースに割り当てられていないジョブのなかから、次にどのジョブをリソースに割り当てるかを、比較キーに基づいて選択する選択手段であり、このジョブ選択部は、比較キーについてあらかじめ決められた同値条件を満たすジョブが複数あるとき、それら複数のジョブの中から次の優先度の比較キーに基づいて割り当てるジョブを選ぶ。



【特許請求の範囲】

【請求項1】 複数のジョブを複数のリソースに割り当てることでスケジュールを作成するスケジュール作成装置において、

互いに優先度の異なる複数の比較キーに基づいて、次のジョブを割り当てるか選ぶ選択手段と、選ばれたジョブをいずれかのリソースに割り当てる手段と、を備え、

前記選択手段は、比較キーについてあらかじめ決められた同値条件を満たすジョブが複数あるとき、それら複数のジョブの中から次の優先度の比較キーに基づいて割り当てるジョブを選ぶように構成されたことを特徴とするスケジュール作成装置。

【請求項2】 どのリソースにどのようなジョブが割り当てられたかに応じて変化する値に基づいて、前記同値条件を設定したことを特徴とする請求項1記載のスケジュール作成装置。

【請求項3】 前記選択手段は、複数の比較キーがあるとき、最初の比較キーについて同値条件を満たすジョブを選び、

2番目以降の比較キーに基づいてジョブを選ぶときは、直前の比較キーについて同値条件を満たすジョブのなかから、次の優先度の比較キーに基づいてジョブを選ぶように構成されたことを特徴とする請求項1又は2記載のスケジュール作成装置。

【請求項4】 どのようなリソースがあるかを表すリソースデータと、

どのようなジョブを割り当てるかを、ジョブごとに対応する複数の属性値で表したジョブデータと、

リソースに割り当てるジョブを順次選ぶときの比較条件を表す比較条件データと、

を使い、

前記比較条件データは、前記複数の属性値のうちどれをどのような優先度で比較キーにするかと、比較キーのうち少なくとも1つについての同値条件と、を表すことを特徴とする請求項1から3のいずれか1つに記載のスケジュール作成装置。

【請求項5】 前記比較キーは、それぞれのジョブの最早開始時刻及び最遅終了時刻であることを特徴とする請求項1から4のいずれか1つに記載のスケジュール作成装置。

【請求項6】 複数のジョブのうちどれをリソースに割り当てるかを、ジョブに関する値を比較キーとしてスケジュールリングのために選択するジョブの選択方法において、

比較キーについて値が異なっても同じ値とみなすための同値条件が決められていることを特徴とするジョブの選択方法。

【請求項7】 複数のジョブを複数のリソースに割り当

てることでスケジュールを作成するスケジュール作成方法において、

互いに優先度の異なる複数の比較キーに基づいて、次のジョブを割り当てるか選ぶ選択のステップと、選ばれたジョブをいずれかのリソースに割り当てるステップと、を含み、

前記選択のステップは、比較キーについてあらかじめ決められた同値条件を満たすジョブが複数あるとき、それら複数のジョブの中から次の優先度の比較キーに基づいて割り当てるジョブを選ぶことを特徴とするスケジュール作成方法。

【請求項8】 どのリソースにどのようなジョブが割り当てられたかに応じて変化する値に基づいて、前記同値条件を設定したことを特徴とする請求項6又は7記載のスケジュール作成方法。

【請求項9】 前記選択のステップは、複数の比較キーがあるとき、

最初の比較キーについて同値条件を満たすジョブを選び、

2番目以降の比較キーに基づいてジョブを選ぶときは、直前の比較キーについて同値条件を満たすジョブのなかから、次の優先度の比較キーに基づいてジョブを選ぶことを特徴とする請求項6から8のいずれか1つに記載のスケジュール作成方法。

【請求項10】 コンピュータを使って、複数のジョブを複数のリソースに割り当てることでスケジュールを作成するスケジュール作成用ソフトウェアを記録した記録媒体において、

そのソフトウェアは前記コンピュータに、

互いに優先度の異なる複数の比較キーに基づいて、次のジョブを割り当てるか選択させ、

選択されたジョブをいずれかのリソースに割り当てさせ、

前記ジョブの選択では、比較キーについてあらかじめ決められた同値条件を満たすジョブが複数あるとき、それら複数のジョブの中から次の優先度の比較キーに基づいて割り当てるジョブを選択させることを特徴とするスケジュール作成用ソフトウェアを記録した記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、コンピュータを使ってスケジュールを作成する技術の改良にかかわるもので、より具体的には、ユーザの意図を直接反映した柔軟なスケジュールリングを行うようにしたものである。

【0002】

【従来の技術】生産計画、プロジェクト計画、時間割作成といったいろいろな計画立案の分野で、コンピュータを使ったスケジュール作成の技術が利用されている。スケジュール作成は、スケジュールリングとも呼び、与えられた条件を満たすようにジョブをリソースに割り当てる

ことである。ここで、ジョブとはスケジューリングを行う対象であり、作業の各工程など、行わなければならない仕事の単位である。また、リソースとは、ジョブを割り当てる対象であり、個々の生産設備や人員のように仕事を処理するために必要な資源である。

【0003】スケジューリングでは、与えられた条件を満たすように、適切なジョブを選んで次々と適切なリソースに割り当てる必要がある。例えば、いくつかの生産設備と、それぞれ納期の決まっている複数のジョブを処理する場合、長時間かかるジョブが一部の生産設備に片寄れば納期割れが発生するし、ある生産設備で、すぐに済んでしまうジョブを処理した後に、ずっと後のある時刻まで開始できないようなジョブが割り当てられれば無駄な空き時間が多くなる。

【0004】すなわち、一般にジョブには、例えばある時間までは開始できないとか、ある納期までに終了しなければならないといった条件、つまり割り当てる期間が設定されている。また、一方で、同じジョブでも、それを処理できるリソース、すなわち割り当てる可能なリソースは複数あり、このようにあるジョブからみて割り当てる期間がいくつかのリソースを「代替リソース」と呼ぶ。そして、ジョブに設定された割り当てる可能な期間のうち、ジョブが最も早く開始できる時刻を「最早開始時刻」、そのジョブに許される最も遅い終了時刻を「最遅終了時刻」と呼ぶ。

【0005】また、リソースに割り当てたジョブがそのリソースを使用する期間を「作業時間」と呼び、最早開始時刻から最遅終了時刻までの期間から作業時間を差し引いた時間を「余裕時間」と呼ぶ。また、ジョブがまだリソースに割り当てられていない状態を「未割当」と呼び、割り当てられたジョブの終了時刻が最遅終了時刻より遅い状態を「納期割れ」、最遅終了時刻からジョブの終了時刻までの期間を「納期遅れ」と呼ぶ。また、スケジューリングが終了した状態において、あるジョブの終了時刻から次に割り当てられたジョブの開始時刻までの期間を「空き時間」と呼ぶ。

【0006】ここで、スケジューリングを行う方法はいくつかあるが、そのうちの一つについて、上に説明したようなそれぞれの用語を前提に説明する。例えば、初期状態では、リソースには何も割り当てられておらず、全てのジョブが未割当である。このような未割当のジョブの中からジョブを一つ選択し、代替リソースの中から割り当てるリソースを決定し、リソースに選択したジョブを割り当てる。そして、このようなジョブの選択と割り当てるという一連の処理を繰り返すことによってスケジューリングを行い、定められた終了条件を満たしたときにスケジューリングを終了する。

【0007】このようなスケジューリングでは、未割当の複数のジョブの中から、次にどのジョブを割り当てるか選ぶとき、ジョブの属性値を比較キーとして未割当の

ジョブ同士を比べ、その場の状況に最適なジョブを選ぶ必要がある。

【0008】例えば、それぞれのジョブが最早開始時刻や余裕時間といった属性値を持っているとき、まず最早開始時刻を比較キーとして使い、最早開始時刻が一番早いジョブを選ぶ。そして、最早開始時刻が全く同じジョブが複数あるときは、それらのジョブの余裕時間を第2の比較キーとしてジョブ同士を比較し、余裕時間が少ないほうのジョブを選んでリソースに割り当てる。

【0009】ここで、いくつかある比較キーを適用する優先順位と、スケジューリングの結果との間には、密接な関係がある。例えば、未割当のジョブの中からジョブを一つ選択するときに、最早開始時刻の早いジョブを選択し、割り当てる可能な時間内でもっとも早く割付可能なリソースに対して、最も早い時刻に割付を行うようにすると、一般的にリソースの空き時間が少なくなり稼働率が高くなる。また、例えば、ジョブの切替時に段取りのための時間（段取り時間）が必要な場合もあるが、この場合は段取り時間が最も短くなるジョブを選択すると稼働率が高くなる。

【0010】

【発明が解決しようとする課題】しかしながら、上に述べたような従来のスケジューリングの技術では、どの比較キーからジョブの比較に使うかという優先順位は固定されていたため、優先順位の内容に応じてスケジューリングの結果が片寄るといった問題があった。ここで、従来のスケジューリングの具体例について説明する。この例では、同一時刻においてそれぞれのリソースに割り当てられるジョブの数は一つまでとし、ジョブを分割したり複数のリソースにまたがって割り当てることはしないこととする。

【0011】そして、まず、第1の例として、割り当てるジョブを選ぶ第一比較キーとして、「最早開始時刻」の「早い順」でジョブを選ぶことにし、また、第二比較キーとしては、「余裕時間」の「少ない順」でジョブを選択し、そのように選んだジョブをリソースに割り当てることでスケジューリングを行う場合について説明する。この場合、スケジューリングの大まかな手順は次のようになる。

(1) 初期状態では全てのジョブが未割当である。

(2) 未割当のジョブの中から最早開始時刻の最も早いジョブを選択する。

このとき、もし最早開始時刻が同じジョブが複数あれば、余裕時間の少ないジョブを選択する。また、未割当のジョブがない場合はスケジューリングを終了する。

【0012】(3) 選択したジョブをリソースに割り当てる。具体的には、代替リソースの中で最も早い時刻にそのジョブを割り当てることのできるリソースを選択し、最も早い時刻に割り当てる。もし割り当てる条件が同じリソースが複数ある場合には、リソースのID番号の若

い順といったように、先に指定されているリソースを選択する。このような割付が済んだ後、(2)へ戻って処理を繰り返す。

【0013】ここで、このような手順を、図4に示すジョブのデータと図5に示すリソースのデータに適用してスケジューリングを行うときの具体的な手順を説明する。この場合、スケジューリングを始めたばかりの時点では、上の手順(1)に述べたように全てのジョブが未割当である。次に上の手順(2)にしたがってジョブの選択を行う。

【0014】つまり、図4に示すジョブのデータでは、最早開始時刻の最も早いジョブはJ-01とJ-02であるが、上の手順(2)にしたがって、余裕時間の少ないJ-02が選択される。このように選択されたジョブJ-02は、上の手順(3)にしたがってリソースに割り当てられる。このジョブJ-02をどのリソースに割り当てるかという点については、J-02の代替リソースにはM2とM3があり、期間を日付で表す場合、どちらも4/1から割当てることが可能であるが、このような場合は代替リソースとして先に指定されているM2に割当てられる。

【0015】この結果、ジョブJ-02の作業時間は3であるので、図12に示すようにリソースM2は4/1から4/3までジョブJ-02のために使用されることになる。このような割当の結果、ジョブJ-02は未割当ではなくなり、処理は上に述べた手順(2)に戻る。

【0016】続いて、残る未割当のジョブの中から同じようにジョブの選択を行うと、今度はジョブJ-01が選択される。このジョブJ-01を割り当てることができる代替リソースとしてはリソースM1とM2があるが、このうちリソースM2にはすでに先程の処理でジョブJ-02が割当てられている。このため、リソースM2については、今回のジョブJ-01を割当可能な時刻は4/3からとなっている。

【0017】一方、リソースM1については4/1から割当可能である。この場合には、早い時刻に割当可能なリソースとしてリソースM1が選択され、ジョブJ-01はリソースM1の4/1から4/5までの期間に割当てられる。以降、同様にジョブの選択と割当を繰り返すことによって、最終的に図13に示すスケジュールが得られる。このように得られたスケジュールでは、空き時間の総計は2日であるが、全てのジョブの納期遅れの総計は37日となる。

【0018】上に述べたスケジューリングの例では、第一比較キーとして、「最早開始時刻」の「早い順」でジョブを選び、第二比較キーとして、「余裕時間」の「少ない順」でジョブを選んだが、次に、第2の例として、これらの第一比較キーと第二比較キーを入れ替えて、第一比較キーを「余裕時間」の「少ない順」で、また、第二比較キーを「最早開始時刻」を「早い順」としてジョ

ブを選択することでスケジューリングを行う例を説明する。

【0019】この第2の例では、処理を始めたばかりの初期状態では、上の手順(1)で述べたように全てのジョブが未割当で、この点は上に説明した第1の例と同じである。しかし、次の手順(2)では、第1の例とは逆の比較キーが使われるため、未割当のジョブの中から余裕時間の少ないジョブJ-03が選択される。このJ-03を割り当てることができる代替リソースはリソースM1とM3であり、これらのリソースM1とM3については、どちらも4/2からジョブを割当てることが可能である。このため、これらのリソースM1とM3のなかからは、代替リソースとして先に指定されているリソースM1がジョブJ-03の割当先のリソースとして選択される。このリソースM1にジョブJ-03を割当てた結果を図6に示す。

【0020】この第2の例では、次に選択されるジョブは未割当ジョブの中でジョブJ-02となり、このジョブJ-02を割り当てることができる代替リソースはリソースM2とM3である。そして、これら2つの代替リソースM2とM3のうちでは、より早い時刻にジョブを配置できる方のリソースM2にジョブJ-02が割当てられる。以降、同様に、第1の例とは違った比較キーの優先順位にしたがって、余裕時間の少ないジョブから選択、割当が行われ、スケジュールが作成される。

【0021】このように、第1の例とは逆に、余裕時間の同じ場合に、最早開始時刻の早いジョブが選択されてリソースに割り当てられる結果、最終的に作られたスケジュールは図14に示すような内容となり、この第2の例では、納期遅れの総計は1日であるが、空き時間の総計は22日になる。

【0022】この結果を上にも述べた第1の例と比較すれば明らかなように、最早開始時刻を第一比較キーにした第1の例では、空き時間は少なくなるが納期遅れが非常に多くなり、一方、余裕時間を第一比較キーにした第2の例では、納期遅れは少なくなるが空き時間が非常に多くなるため、納期遅れと空き時間の両方を適度に減らすことはできない。このように、従来技術では、スケジューリングの処理を行う間、比較キーの順序を終始一貫した順序に固定してジョブの選択を行っていたため、複数の基準からみてバランスの良いスケジュールを得ることは困難であった。

【0023】なお、スケジューリングでのジョブの選択について、ファジー推論を利用したスケジューリング(特開平8-118210)では、ジョブの複数の属性を入力し、ファジー推論によりジョブを選択するための優先度を決定し、それに基づいてジョブの選択を行うことで、複数の属性を考慮した選択を行う技術が提案されている。

【0024】しかし、このような技術を使う場合も、推

論に使うメンバーシップ関数やパラメータなどの調整が難しいだけでなく、比較キーの優先順序は予め決定され、状況に応じて動的に変えられないため、スケジューリングの手順や結果にユーザの意図を直接反映させることができないという問題があった。

【0025】つまり、このようなファジー推論を使っても、複数の属性を考慮する考慮の仕方やバランスが固定されているため、例えば、最早開始時刻の違いが2, 3日であれば余裕時間の少ないジョブを優先する、といった非線形の評価基準に基づいた臨機応変な優先度の処理を実現することはできなかった。そして、このようなファジーを使った手法でも、割付を行う前に全てのジョブに対して優先度を固定してしまうため、リソースの負荷など途中のスケジューリング状況に応じて優先度を変えることができないという問題があった。

【0026】この発明は、上に述べたような従来技術の問題点を解決するために提案されたもので、その目的は、ユーザの意図を直接反映したスケジューリングの技術を提供することである。また、本発明の他の目的は、状況に応じて優先度を動的に変更することで、柔軟なスケジューリングを行う技術を提供することである。

【0027】

【課題を解決するための手段】上記の目的を達成するため、請求項1の発明は、複数のジョブを複数のリソースに割り当てることでスケジュールを作成するスケジュール作成装置において、互いに優先度の異なる複数の比較キーに基づいて、次にどのジョブを割り当てるか選ぶ選択手段と、選ばれたジョブをいずれかのリソースに割り当てる手段と、を備え、前記選択手段は、比較キーについてあらかじめ決められた同値条件を満たすジョブが複数あるとき、それら複数のジョブの中から次の優先度の比較キーに基づいて割り当てるジョブを選ぶように構成されたことを特徴とする。請求項7の発明は、請求項1の発明を方法という見方からとらえたもので、複数のジョブを複数のリソースに割り当てることでスケジュールを作成するスケジュール作成方法において、互いに優先度の異なる複数の比較キーに基づいて、次にどのジョブを割り当てるか選ぶ選択のステップと、選ばれたジョブをいずれかのリソースに割り当てるステップと、を含み、前記選択のステップは、比較キーについてあらかじめ決められた同値条件を満たすジョブが複数あるとき、それら複数のジョブの中から次の優先度の比較キーに基づいて割り当てるジョブを選ぶことを特徴とする。請求項10の発明は、請求項1, 7の発明を、コンピュータのプログラムを記録した記録媒体という見方からとらえたもので、コンピュータを使って、複数のジョブを複数のリソースに割り当てることでスケジュールを作成するスケジュール作成用ソフトウェアを記録した記録媒体において、そのソフトウェアは前記コンピュータに、互いに優先度の異なる複数の比較キーに基づいて、次にどの

ジョブを割り当てるか選択させ、選択されたジョブをいずれかのリソースに割り当てさせ、前記ジョブの選択では、比較キーについてあらかじめ決められた同値条件を満たすジョブが複数あるとき、それら複数のジョブの中から次の優先度の比較キーに基づいて割り当てるジョブを選択させることを特徴とする。請求項6の発明は、複数のジョブのうちどれをリソースに割り当てるかを、ジョブに関する値を比較キーとしてスケジューリングのために選択するジョブの選択方法において、比較キーについて値が異なっても同じ値とみなすための同値条件が決められていることを特徴とする。請求項1, 7, 10, 6の発明では、ジョブを選ぶための比較キーごとに、同一視してよい程度の差の範囲を同値条件として設定しておく。そして、ある優先度の比較キーで比べた結果、同値条件を満たすほど差が少ないジョブ同士については、次の優先度の比較キーによって選ばれることで、柔軟なスケジューリングが実現される。また、個々の比較キーについて、どの程度の差であれば同一視してよいかを同値条件として設定することで、ユーザの意図がスケジューリングに直接反映される。例えば、第一比較キーを最早開始時刻、同値条件を「最早開始時刻の差が3以内」とし、第二比較キーを余裕時間とした例を考える。二つのジョブを比較した時に最早開始時刻の差が3より大きければ最早開始時刻の早いジョブが選択され、最早開始時刻の差が3以内の時には同値条件により同じ値であるとみなされ、次の比較キーである余裕時間の少ないジョブが選択されることになる。これにより、ユーザの意図を直接反映した柔軟なスケジューリングが可能になる。

【0028】請求項4の発明は、請求項1から3のいずれか1つに記載のスケジュール作成装置において、どのようなリソースがあるかを表すリソースデータと、どのようなジョブを割り当てるかを、ジョブごとに対応する複数の属性値で表したジョブデータと、リソースに割り当てるジョブを順次選ぶときの比較条件を表す比較条件データと、を使い、前記比較条件データは、前記複数の属性値のうちどれをどのような優先度で比較キーにするかと、比較キーのうち少なくとも1つについての同値条件と、を表すことを特徴とする。請求項4の発明では、ジョブの持っている属性値のうちどれとどれを比較キーとし、また、比較キーのどれについてどのような条件が満たされれば次の優先度の比較キーでジョブを選んでよいかについて、比較条件データを書き換えることで容易に制御することができる。このため、ユーザの意図をスケジューリングに反映させたり、いろいろな分野にこの発明を適用することが容易になる。

【0029】請求項5の発明は、請求項1から4のいずれか1つに記載のスケジュール作成装置において、前記比較キーは、それぞれのジョブの最早開始時刻及び最遅終了時刻であることを特徴とする。請求項5の発明は、

ジョブの実行がいつから可能にかかわる最早開始時刻や、納期に直結する最遅終了時刻を比較キーとして使うので、スケジューリングにかかわる多くの分野に効果的に適用することができる。

【0030】請求項2の発明は、請求項1記載のスケジュール作成装置において、どのリソースにどのようなジョブが割り当てられたかに応じて変化する値に基づいて、前記同値条件を設定したことを特徴とする。請求項8の発明は、請求項2の発明を方法という見方からとらえたもので、請求項6又は7記載のスケジュール作成方法において、どのリソースにどのようなジョブが割り当てられたかに応じて変化する値に基づいて、前記同値条件を設定したことを特徴とする。請求項2、8の発明では、スケジューリングが途中で進まない決まらないような値を使って同値条件が設定できるので、スケジューリングの状況に応じてより柔軟なスケジューリングを行うことができる。例えば、連続する2つのジョブの種類の組み合わせに応じて2つのジョブの間の段取り時間は変化するもので、それにとまってそれらのジョブの終了時刻も変化する。請求項2、8の発明によれば、同値条件を、このように変動する終了時刻と、候補となる各ジョブの最早開始時刻との時間間隔などに基づいて設定することができる。

【0031】請求項3の発明は、請求項1又は2記載のスケジュール作成装置において、前記選択手段は、複数の比較キーがあるとき、最初の比較キーについて同値条件を満たすジョブを選び、2番目以降の比較キーに基づいてジョブを選ぶときは、直前の比較キーについて同値条件を満たすジョブのなかから、次の優先度の比較キーに基づいてジョブを選ぶように構成されたことを特徴とする。請求項9の発明は、請求項3の発明を方法という見方からとらえたもので、請求項6から8のいずれか1つに記載のスケジュール作成方法において、前記選択のステップは、複数の比較キーがあるとき、最初の比較キーについて同値条件を満たすジョブを選び、2番目以降の比較キーに基づいてジョブを選ぶときは、直前の比較キーについて同値条件を満たすジョブのなかから、次の優先度の比較キーに基づいてジョブを選ぶことを特徴とする。請求項3、9の発明では、複数の比較キーを使って最適なジョブを順次選んでゆくと、1つ前の優先度の比較キーに基づいて同値条件の範囲とされるジョブだけが次の比較キーで比較される候補となる。このため、個々の比較キーに基づいて最適なジョブを次々と選んだ結果、最初の比較キーでいえば最適なものからかけ離れたジョブが最終的に選ばれる、という問題を避けることができる。なお、「最適なジョブ」は比較キーの内容や形式に応じて違ってくるが、例えば比較キーが昇順の数値なら最小値、降順の数値なら最大値に対応するジョブが最適なジョブである。

【0032】

【発明の実施の形態】以下、本発明の実施の形態（以下「実施形態」という）について図面を参照しながら説明する。なお、本発明は、周辺機器を持つコンピュータを、ソフトウェアで制御することによって実現されることが一般的と考えられる。この場合、そのソフトウェアは、この明細書の記載にしたがった命令を組み合わせることで作られ、上に述べた従来技術と共通の部分には従来技術で説明した手法も使われる。また、そのソフトウェアは、プログラムコードだけでなく、プログラムコードの実行のときに使うために予め用意されたデータも含む。

【0033】〔1. 第1実施形態〕

〔1-1. 第1実施形態の構成〕ここで、図1は、本発明の第1実施形態の構成を示す機能ブロック図であり、上に述べたようなソフトウェアSの動作が、コンピュータCによって実現されることを概念的に示すものである。すなわち、コンピュータCは、CPU(C1)と、RAMなどを使ったメインメモリC2と、ハードディスク装置などの補助記憶装置C3と、キーボードやマウスなどの入力装置C4と、CRTディスプレイやプリンタなどの出力装置C5といったハードウェアを備えていて、上に説明したようなソフトウェアSは、これらハードウェアのような物理的な資源を活用することで本発明の作用効果を実現する。

【0034】但し、本発明を実現する具体的なソフトウェアやハードウェアの構成はいろいろ変更することができる。例えば、ソフトウェアの形式には、コンパイル、インタプリタ、アセンブラなどいろいろあり、外部との情報をやり取りするにも、フロッピーディスクなどの着脱可能な記録媒体、ネットワーク接続装置などいろいろ考えられる。また、本発明を実現するソフトウェアやプログラムを記録したCD-ROMのような記録媒体は、単独でも本発明の一態様である。さらに、本発明の機能の一部をLSIなどの物理的な電子回路で実現することも可能である。

【0035】以上のように、コンピュータを使って本発明を実現する態様はいろいろ考えられるので、以下では、本発明や実施形態に含まれる個々の機能を実現する仮想的回路ブロックを使って、本発明と実施形態とを説明する。

【0036】すなわち、第1実施形態は、複数のジョブを複数のリソースに割り当てることでスケジュールを作成するスケジュール作成装置であり、第1実施形態におけるソフトウェアSでは、スケジューリングの対象としてどのようなジョブがあり、また、どのようなリソースがあるかを、それぞれジョブデータD1及びリソースデータD2として与えられる。

【0037】このうちジョブデータD1には、それぞれのジョブについて、名前、代替リソース、最早開始時刻、最遅終了時刻、作業時間など属性のデータが保持さ

れている。また、リソースデータD2には、それぞれのリソースの能力、稼働時間、割り当てられたジョブなど、リソースに関するデータが保持される。また、第1実施形態では、次にリソースに割り当てるジョブを選ぶときにジョブのどの属性を比較キーとしてジョブ同士を比較するかを、比較条件データD3として与える。

【0038】この比較条件データは、ジョブの比較を行うときの比較条件、すなわち、ジョブごとに複数存在する属性値のうち、どれとどれをどのような優先順位で比較キーにするかと、比較キーのうち少なくとも1つについての同値条件と、を含んでいる。ここで、同値条件とは、比較キーについて、ジョブの属性値が異なっているも同じ値とみなすための条件である。

【0039】また、ソフトウェアSは、同値条件設定部1と、優先度比較部2と、ジョブ選択部3と、スケジューリング処理部4と、を備えている。このうち同値条件設定部1は、比較条件データD3を入力することで設定する部分である。また、優先度比較部2は、比較条件データD3に含まれる複数の比較キーの優先度を互いに比較することで、個々の比較キーをジョブの選択の基準として使う順序を決める部分である。

【0040】また、ジョブ選択部3は、選択対象となっているジョブ、すなわちまだリソースに割り当てられていないジョブのなかから、次にどのジョブをリソースに割り当てるかを、比較キーに基づいて選択する選択手段であり、このジョブ選択部は、比較キーについてあらかじめ決められた同値条件を満たすジョブが複数あるとき、それら複数のジョブの中から次の優先度の比較キーに基づいて割り当てるジョブを選ぶように構成されている。

【0041】また、スケジューリング処理部4は、このように選択されたジョブをいずれかのリソースに割り当てる処理を行うことでスケジューリングの作成を行う手段である。また、第1実施形態における比較キーは、それぞれのジョブの最早開始時刻及び最遅終了時刻であるものとする。

【0042】〔1-2. 第1実施形態の作用〕上に述べたような構成を備えた第1実施形態は、次のように作用する。

〔1-2-1. スケジューリング全体の手順〕第1実施形態では、比較条件データに表された比較キーと同値条件とを使ってジョブを選択するが、このようなジョブの選択は、次に例示するようなスケジューリング全体の手順の一部として行われる。すなわち、スケジューリング作成手順を大まかに示すと次のようになる。

(1) まず、ユーザが、入力装置C4などを使ってジョブデータD1やリソースデータD2を入力することによって設定する。同様に、比較条件データD3についても同値条件設定部1を通して設定する。

(2) そして、ジョブ選択部3が、選択対象となってい

るジョブの中からジョブを一つ選択する。なお、このとき選択可能なジョブがなくなっていた場合は、全てのジョブについてリソースへの割付けが終了している状態であるから、スケジューリングを終了する。

(3) 続いて、スケジューリング処理部4が、選択されたジョブを割り当てるリソースを決定し、決定されたリソースにジョブを割当て、上に述べた(2)におけるジョブの選択へ戻る。

【0043】なお、選択したジョブをスケジューリング処理部4がどのような手法や基準でどのリソースに割り当てるかは自由であり、例えば、いわゆる山積み、山崩しによるスケジューリングの場合には全体の手順は次のようになる。

(1) まず、個々のリソースの容量を考慮せずに、全てのジョブを一旦各リソースに割当てる。

(2) 次に、ジョブ選択部3が、リソースの容量以上に割当てられたジョブの中からジョブを一つ選択する。なお、このとき選択可能なジョブがなくなっている場合は、どのジョブもリソースの容量内に収まっている状態であるから、スケジューリングを終了する。

(3) そして、スケジューリング処理部4が、選択されたジョブをリソースの容量を越えない時刻まで移動させたうえ、上に述べた(2)におけるジョブの選択へ戻る。

【0044】このように、本実施形態は特にジョブの選択に特徴があるもので、選択されたジョブをどのような手法でどのように処理してスケジューリングを作成してもよい。このため、以下では、ジョブ選択部3によるジョブの選択に重点をおいて本実施形態の作用や他の実施形態などについて説明する。

【0045】〔1-2-2. ジョブ選択の手順〕すなわち、ジョブ選択部3へは、ジョブデータD1のうち選択対象となっているジョブのデータが入力され、また、比較条件データD3に含まれる比較キー及び同値条件のうち優先順位の高いものから順に渡される。そして、ジョブ選択部3は、それぞれの時点で、選択対象として残っているジョブを2つ取り出し、比較して残ったほうをさらに別のジョブと比較するという二項比較を行う。このように、ジョブ選択部3がジョブ同士の二項比較によってジョブを選択する手順の例を図2に示す。

【0046】〔1-2-3. ジョブ選択の例〕また、図3に示す比較条件データの例と図4に示すジョブデータの例を使ってジョブの選択を行う手順をこの図2に沿って説明する。なお、リソースデータは、図5に示すような内容のものが与えられているものとする。まず、ジョブJ-01とJ-03とを比較する場合、次のような二項比較が行われる。

【0047】すなわち、比較する2つのジョブJ-01とJ-03をそれぞれ、変数AとBにセットし、また、変数nを0とする(ステップ1)。続いて、nの値を1

増やし、すなわちインクリメントする(ステップ2)。最初にこのステップが実行される時、0だったnが1となる。ここで、比較条件データを見るとn番目、例えば1番目のキーが存在するので(ステップ3)、このn番目の比較キー「最早開始時刻」と、それに対応する同値条件「最早開始時刻の差が3以内」を、予め決められた変数やワークエリアなどにセットする(ステップ4)。

【0048】そして、このようにセットされたn番目の比較キーとその同値条件を使って、現在A、BにそれぞれセットされているジョブJ-01とJ-03に関するチェックを行う。具体的には、まず、ジョブJ-01、J-03の最早開始時刻はそれぞれ、4/1、4/2であり、両者の差は1日であるからここでは同値条件を満たし、処理はステップ2に戻る。

【0049】ステップ2では、nの値を再び一つ増やし、nを2とする。そして、比較条件データD3にはn番目、このときは2番目の比較キーが存在するので(ステップ3)、この比較キー「余裕時間」を上記述べたワークエリアなどにセットする(ステップ4)。なお、この比較キーには同値条件は設定されていないので、このときは同値条件は「無し」となる。この場合、続く同値条件のチェック(ステップ5)では、同値条件がないので同値条件は満たされなかったと判断される。

【0050】このため、現在の比較キーである余裕時間について、AにセットされているジョブJ-01と、BにセットされているJ-03を大小比較する(ステップ6)。この場合、この比較キー「余裕時間」については、順序が昇順と設定されているので、すなわち余裕時間の少ないものほど先に選択すると決められているので、余裕時間の少ないジョブJ-03(B)が選択され、このジョブJ-03がスケジュール処理部4に対して出力される。

【0051】〔1-2-4. ジョブ選択の他の例〕また、同様にジョブJ-01とJ-10の比較を行った場合は次のようになる。すなわち、まず、これらジョブJ-01とJ-10をそれぞれ、AとBにセットし、nを0とする(ステップ1)。続いて、nの値を増やし、nを1とする(ステップ2)。ここで、比較条件データを見ると1番目のキーは存在するので(ステップ3)比較キー、同値条件についてそれぞれ「最早開始時刻」、「最早開始時刻の差が3以内」にセットする(ステップ4)。

【0052】ここで、まず、同値条件のチェックを行う(ステップ5)。このときは、ジョブJ-01とJ-10の最早開始時刻はそれぞれ、4/1と4/20であり、差が19となるので同値条件を満たさない。そこで、次に、比較キーである最早開始時刻で比較する(ステップ6)。この判断では、順序が昇順と設定されているので、最早開始時刻が早いジョブJ-01が選択され

る。

【0053】〔1-2-5. スケジューリングの例〕以上のようなジョブ選択を使ったスケジューリングの例について、図3の比較条件データと図4のジョブのデータを使って説明する。まず、スケジューリングを始める時点では、初めは全てのジョブが未割当である。

【0054】そして、同値条件「最早開始時刻の差が3以内」を満たし、かつ第一比較キーである最早開始時刻で昇順に並べ、そのなかから同値条件を満たすものを選ぶと、ジョブJ-01、J-02、J-03、J-04、J-05が選択の候補となる。さらにこの中で第二比較キーによる比較を行う。この第二比較キーについては同値条件が設定されていないので、余裕時間の最も少ないジョブであるジョブJ-03が選択される。そこで、まずこのジョブJ-03がリソースM1に割り当てられると図6に示すスケジュールの状況となる。

【0055】続いて、次に割り当てるジョブの選択を行うが、第一比較キーによる残る候補はジョブJ-01、J-02、J-04、J-05であり、上に説明したのと同じように、第二比較キーに基づいて、この中で余裕時間の最も少ないジョブJ-02が選択されM2に割り当てられる。

【0056】同様に、第1比較キーの同値条件を満たすジョブが、第2比較キーを基準としてジョブJ-05、J-04、J-01の順で選択されリソースに割り当てられる。そして、このうち最後のジョブJ-01がリソースに割り当てられた時点では、未割当のジョブの中で最早開始時刻が最も早いジョブはジョブJ-06であり、その最早開始時刻は4/5となる。

【0057】したがって第一比較キーと同値条件からこのJ-06に加え、同値条件を満たすジョブJ-07、J-08、J-09が選択の候補となり、余裕時間の最も少ないのはジョブJ-08とJ-09であるが、これら2つのジョブJ-08とJ-09の間では第2比較キーである余裕時間に違いがなく、これ以上別の比較キーが設定されていないので、このような場合はデータを読み込んだ順に選択され、リソースへの割当が行われる。

【0058】以降、同様にスケジューリングを行うと図7に示すスケジュールが得られ、納期遅れの総計は14日、空き時間の総計は6日となる。ここで、このような本実施形態によるスケジューリング結果を、前述の従来手法すなわち従来技術と比較した結果を図8に示す。すなわち、この図に示すように、従来の手法では納期遅れと空き時間のどちらか一方は少ないが、どちらか一方は多いというスケジューリング結果になっているのと比較して、本実施形態では、両者のバランスが取れたスケジュール結果が得られている。

【0059】〔1-3. 第1実施形態の効果〕以上に説明したように、第1実施形態では、ジョブを選ぶための

比較キーごとに、同一視してよい程度の差の範囲を同値条件として、比較条件データD3に設定しておく。そして、ジョブ選択部3において、ある優先度の比較キーで比べた結果、同値条件を満たすほど差が少ないジョブ同士については、次の優先度の比較キーによって選ばれることで、柔軟なスケジューリングが実現される。また、個々の比較キーについて、どの程度の差であれば同一視してよいかを同値条件として設定することで、ユーザの意図がスケジューリングに直接反映される。すなわち、ユーザの意図にしたがってスケジューリングの内容を調整したいとき、従来のように各種パラメータを調整するよりも、調整の内容や意義がユーザにも理解しやすくなる。

【0060】また、第1実施形態では、ジョブの持っている属性値のうちどれとどれを比較キーとし、また、比較キーのどれについてどのような条件が満たされれば次の優先度の比較キーでジョブを選んでよいかについて、比較条件データD3を書き換えることで容易に制御することができる。このため、ユーザの意図をスケジューリングに反映させたり、いろいろな分野にこの発明を適用することが容易になる。

【0061】また、第1実施形態では、ジョブの実行がいつから可能にかかわる最早開始時刻や、納期に直結する最遅終了時刻を比較キーとして使うので、スケジューリングにかかわる多くの分野に効果的に適用することができる。

【0062】〔2. 第2実施形態〕次に、第2実施形態は、第1実施形態と同様の構成(図1)に対して、同値条件として、スケジューリング途中で動的に変化する値を組み込むことによって、ジョブの効果的な選択が可能になる例である。

【0063】すなわち、あるリソースにおいて二つのジョブを連続して割当てる場合に、二つのジョブの組み合わせによっては、ジョブの正味の作業時間に含まれない準備のための時間を必要とする場合がある。この準備のための時間を段取り時間と呼ぶ。例えば、機械加工の場合には工具の交換やセットアップのための時間が段取り時間であり、連続する二つのジョブの組み合わせによってこの段取り時間は変化する。

【0064】このような前提で、例えば、時刻の早い方からジョブの割当を行って、図9に示すような状況だとする。この図において、網掛け部分は段取り時間を表し、また、矢印は、全てのリソースのうちで最後のジョブの終了時刻が一番早い時刻を示しており、この時刻を最早割当時刻と呼ぶ。この最早割当時刻はどのリソースにどのジョブが割り当てられたかに応じて、スケジューリングの途中で動的に変化する値である。

【0065】この状況に対して、従来技術において、比較キーの優先順位は、最早開始時刻、段取り時間の順で指定されている場合を考える。また、この例では、二つ

のジョブJ-AとJ-Bがあり、最早開始時刻、段取り時間がそれぞれジョブJ-Aは4/1, 10, ジョブJ-Bは4/5, 5であるとし、最早割当時刻が4/10であるとする。

【0066】この場合、ジョブJ-AとJ-Bの最早開始時刻はどちらも最早割当時刻より早いので、実際はどちらのジョブでも空き時間を作ることなく割当が可能であるが、最早開始時刻や段取り時刻といった固定された値だけを比較キーに使う場合、第一比較キーが最早開始時刻であるために、段取り時間がより大きいジョブJ-Aが選択されるため、効果的なスケジューリングが困難となる。

【0067】また、同様にジョブJ-AとJ-Bがあり、比較キーの優先順位が上の例とは逆に、段取り時間、最早開始時刻の順で指定されていたとすると、段取り時間が少ないジョブJ-Bの方がジョブJ-Aよりも先に選択される。しかし、例えば図9に示した最早割当時刻が4/10ではなく3/10だったような場合を考えると、先に選択されるジョブJ-Bの最早開始時刻である4/1まで大幅な空き時間が生じる。そして、この空き時間はジョブJ-Bの段取り時間よりもはるかに長いいため、全体として、無駄な空き時間が増大する。したがって、最早割当時刻が3/10であるような場合は、最早開始時刻の早いジョブJ-Aを選択する方が望ましい。

【0068】すなわち、第2実施形態は、同値条件を、どのリソースにどのようなジョブが割り当てられたかに応じて、スケジューリング中に動的に変化する値に基づいて設定する例である。具体的には、上に説明したような図9の状況で、次に割当てるジョブを選択する際、第2実施形態では、図10に示すような比較条件データを予め設定しておく。

【0069】この場合、第一比較キーの同値条件より、最早開始時刻が(最早割当時刻+5)よりも早いジョブはすべて同じとみなされる。すなわち、このようなジョブは、リソースに割当を行う場合に空き時間を作る可能性が少なく、第二比較キーによりその中で段取り時間の最も少ないジョブを選択することができる。

【0070】すなわち、第2実施形態では、このような同値条件が設定されている結果、すべてのジョブが(最早割当時刻+5)よりも遅い場合は、そのなかで最早開始時刻の最も早いジョブが選択されることになる。つまり、すべてのジョブが(最早割当時刻+5)よりも遅い場合は、段取り時間以上の空き時間ができてしまうことが多いので、できるだけ最早開始時刻の早いジョブを選択する方がよい。

【0071】以上のように、第2実施形態では、スケジューリングが途中まで進まないし決まらないような値を使って同値条件が設定できるので、スケジューリングの状況に応じてより柔軟なスケジューリングを行うことが

できる。

【0072】〔3. 第3実施形態〕第3実施形態は、比較キーの同値条件と、次の比較キーを組み合わせること、効果的にスケジューリングを行う例であり、第1実施形態に示したようにジョブを二つ選択して二項比較を行うだけではジョブの選択が適切に行われないことがある問題を解決したものである。

【0073】すなわち、第1実施形態では、複数の比較キーを使って最適なジョブを順次選んでゆくと、1つ前の優先度の比較キーに基づいて同値条件の範囲とされるジョブだけが次の比較キーで比較される候補となる。このため、個々の比較キーに基づいて最適なジョブを次々と選んだ結果、最初の比較キーでいえば最適なものからかけ離れたジョブが最終的に選ばれる場合も考えられる。

【0074】ここで、このような具体例を、図11に示すジョブのデータと図3に示す比較条件データを使ってスケジューリングを行う場合を例にとって説明する。この例では、初めにジョブJ-01とJ-02を比較すると最早開始時刻がそれぞれ4/1、4/4でありその差が3となっている。この場合、これら2つのジョブJ-01とJ-02は、第一比較キーの同値条件を満たすので、第二比較キーの余裕時間で比較を行う結果、ジョブJ-02が選択されることになる。

【0075】次に、選択されたこのジョブJ-02と、まだ割当の行われていない次の候補であるジョブJ-03の比較を行うと、両者は最早開始時刻の差が3なので第一比較キーの同値条件を満たし、第二比較キーの余裕時間で比較を行う結果、今度はジョブJ-03が選択されることになる。同様に、このジョブJ-03とさらにジョブJ-04を比較するとジョブJ-04が選択され、さらに、ジョブJ-04とJ-05を比較するとジョブJ-05が選択される。

【0076】しかし、このジョブJ-05とJ-01を比較すると、両者の最早開始時刻の差が12なので同値条件を満たさず、結局ジョブJ-01が選択される。しかし、この一連の手続きの最初で示したように、この場合は実際にはジョブJ-02を選択することが望ましい。

【0077】このように、二つのジョブを比較していくだけではジョブを比較する順序によって選択されるジョブが異なることになり、特に、比較を続ける結果、優先度の高い基準から大幅に外れ、真に望ましいジョブが選択されないことがある。そこで、第3実施形態では、図1に示したジョブ選択部3において、複数の比較キーがあるとき、最初の比較キーについて同値条件を満たすジョブを選び、2番目以降の比較キーに基づいてジョブを選ぶときは、直前の比較キーについて同値条件を満たすジョブのなかから、次の優先度の比較キーに基づいてジョブを選ぶものである。

【0078】具体的には、この第3実施形態では、次のような手順でジョブの選択を行うことが考えられる。

手順1. まず、nを1にセットする。

手順2. また、ジョブのデータを走査して1番目の比較キーにより昇順の場合は最小値、降順の場合は最大値を検索する。その最小値又は最大値と、そのような最小値又は最大値を持っていたジョブを変数M、Jにセットする。同時に、全てのジョブに選択フラグを立てる。

手順3. 次のn+1番目の比較キーが無い場合はJを選択したジョブとして出力し、処理を終了する。それ以外の場合は次の手順4へ進む。

手順4. 選択フラグの立っているジョブだけを走査し、Mを使ってn番目の同値条件を満たす場合のみ選択フラグを立て、同値条件を満たさない場合はフラグを立てずにリセットする。同時に、同値条件を満たしたジョブの中でn+1番目の比較キーにより昇順の場合は最小値を、降順の場合は最大値をM'にセットしそのような最小値又は最大値を持っていたジョブをJにセットする。

【0079】手順5. nの値をひとつ増やし、MにM'の値をセットし手順3.に戻る。

【0080】すなわち、簡単な例を示せば、ジョブ1, 2, 3, 4, 5を、第1比較キーの条件を満足する順にしたがってソートすることで、ジョブ2, 3, 4, 1, 5という順序が得られ、その中でジョブ2, 3, 4を同値条件に基づいて同値とする。続いて、これらジョブ2, 3, 4を第2比較キーに基づいてチェックし、その結果をソートすることでジョブ3, 2, 4という順序が得られ、このうちジョブ3, 2を同値とする。さらに、これらジョブ3, 2を第3比較キーに基づいてチェックし、その結果をソートすることでジョブ2, 3という順序が得られ、ジョブ2が選択される。

【0081】このような処理を実現するには、具体的には、まず、第1比較キーでそれぞれのジョブの値を検索する処理を行い、このように個々のジョブの値にしたがってソートするときに第2比較キーによる値をチェックし、同様に、第2比較キーの値にしたがってソートするときに、第3比較キーの値をチェックする、という具合の手順を使うことで、少ない比較処理で最適なものを選択することができる。

【0082】次に、図4に示したジョブデータと図3に示した比較条件データを使って具体的に説明すると次のようになる。

・まず、上に説明した手順1. よりnを1とする。

・次に、手順2. より、一番目の比較キーが最早開始時刻であり、この比較キーは昇順であるから、最早開始時刻が最小となるジョブとその値を検索する。同時に全てのジョブに選択フラグを立てる。このように走査した結果、Mには4/1がセットされJにはJ-01がセットされる。

・また、手順3. より二番目の比較キーが存在するので

手順4. に進む。

・この手順4. では、今の段階では全てのジョブに選択フラグが立っているので全てのジョブを走査する。このとき、Mの値は4/1であり、同値条件が「最早開始時刻の差が3以内」であるから、選択フラグが立てられるジョブはJ-01, J-02, J-03, J-04, J-05だけになる。そして、2番目の比較キーが余裕時間であり、この比較キーは昇順である。このため、ここで選択フラグが立てられたジョブの中で余裕時間が最小値は0であることから、M'には0がセットされ、JにはJ-03がセットされる。

・次に、手順5. よりnを一つ増やして2としMを0とする。

・そして、今度は手順3. より三番目の比較キーが存在しないのでJにセットされているJ-03を選択したジョブとして出力し処理を終了する。

【0083】このような第3実施形態によれば、ジョブの二項比較を繰り返すことで、優先度の高い基準から大幅にずれたジョブが選択されるという問題を避けることができる。

【0084】〔4. 他の実施形態〕なお、本発明は上記のような各実施形態に限定されるものではなく、次に例示するような他の実施形態も含むものである。例えば、本発明を適用する具体的な分野は自由であり、ジョブやリソースの種類、数、比較キーや属性値の内容、選択したジョブをリソースに割り当てる手法、同値条件の形式などは、具体的な適用課題に応じて自由に定めることができる。

【0085】

【発明の効果】以上説明したように、本発明によれば、ユーザの意図を直接反映したスケジューリングが可能となる。

【図面の簡単な説明】

【図1】本発明の第1実施形態の構成を示す機能ブロック図。

【図2】本発明の第1実施形態における二項比較の手順を示すフローチャート。

【図3】本発明の実施形態における比較条件データの例を示す図。

【図3】

| 比較キー | 順序 | 同値条件 |
|----------|----|--------------|
| 1 最早開始時刻 | 昇順 | 最早開始時刻の差が3以内 |
| 2 余裕時間 | 昇順 | |

【図4】本発明の実施形態におけるジョブデータの例を示す図。

【図5】本発明の実施形態におけるリソースデータの例を示す図。

【図6】本発明の第1実施形態におけるスケジューリングの状況を示す図。

【図7】本発明の第1実施形態におけるスケジューリング結果を示す図。

【図8】本発明の第1実施形態によるスケジューリング結果と、従来技術によるスケジューリング結果を比較する図。

【図9】本発明の第2実施形態にかかわるスケジューリングの状況を示す図。

【図10】本発明の第2実施形態における比較条件データを示す図。

【図11】本発明の第3実施形態におけるジョブデータの例を示す図。

【図12】従来のスケジューリングの状態を例示する図。

【図13】従来のスケジューリングの状態を例示する図。

【図14】従来のスケジューリングの状態を例示する図。

【符号の説明】

- C…コンピュータ
- S…ソフトウェア
- C1…CPU
- C2…メインメモリ
- C3…補助記憶装置
- C4…入力装置
- C5…出力装置
- D1…ジョブデータ
- D2…リソースデータ
- D3…比較条件データ
- 1…同値条件設定部
- 2…優先度比較部
- 3…ジョブ選択部
- 4…スケジュール処理部

【図5】

| リソース名 |
|-------|
| M1 |
| M2 |
| M3 |

【図9】

